

# Density Functional Theory-Based Nanostructure Investigation

Adrian Kopacz

Office of Science, Science Undergraduate Laboratory Internship (SULI)

Northwestern University

Argonne National Laboratory

Argonne, Illinois

August 18, 2005

Prepared in partial fulfillment of the requirements of the Office of Science, Department of Energy's Science Undergraduate Laboratory Internship under the direction of Mihai Anitescu<sup>1</sup> and Dan Negrut<sup>2</sup> in the Mathematics and Computer Science Division at Argonne National Laboratory.

Participant: \_\_\_\_\_

Research Mentor: \_\_\_\_\_

<sup>1</sup>Math. and Comp. Science Division, Argonne National Laboratory; anitescu@mcs.anl.gov

<sup>2</sup>Math. and Comp. Science Division, Argonne National Laboratory; negrut@mcs.anl.gov

## **ABSTRACT**

Density Functional Theory-Based Nanostructure Investigation. ADRIAN KOPACZ (Northwestern University, Evanston IL 60208) MIHAI ANITESCU (Mathematics and Computer Science Division, Argonne IL 60439).

The development of software for the investigation of chemical and mechanical properties of nanostructures promises to elucidate phenomena not observed in bulk materials. The method formulates a two-step approach to compute the electronic density distribution in and around a nanostructure and then the displacement of its nuclei. The Electronic Problem employs interpolation and coupled cross-domain optimization techniques through a process called electronic reconstruction. The Ionic Problem, within a quasicontinuum framework, relocates the nuclei of the nanostructure given the electronic density in the domain. The goal of this work is to implement an object-oriented framework that will provide testing mechanisms of the evolving code. Future work will focus on further enhancements to substantially increase the dimension of the nanostructures that can be simulated by using approaches that include accurate density functional theory (DFT) computation.

## INTRODUCTION

A typical nanostructure has dimensions in the range of 1~100nm. The electronic structure of such materials undergoes drastic changes at reduced dimensions. Their surface to volume ratio is relatively high resulting in new phenomena that are not observed in bulk materials where dimensions are on the order of microns. Nanoscale oxide structures illustrate new regimes of behavior in chemical reactivity [13], magnetic properties [14], charge transport [15], and optical properties [16, 17]. In order to better understand them, it is essential to take a look at the electronic composition of these nanostructures.

To attain properties of a nanostructure, two important quantities must be calculated; the ground state energy,  $\varepsilon_0$  and the ground state electron density,  $\rho(r)$ . These two quantities can be obtained after solving the Schrödinger's equation (1a), an approach that scales very unfavorably with the dimension of the problem.

$$\mathbf{H}\Psi = \varepsilon\Psi \quad (1a)$$

$$\Psi(\mathbf{r}_1, \dots, \mathbf{r}_N) \quad (1b)$$

$$\mathbf{H} = -\sum_{i=1}^N \frac{1}{2} \nabla_i^2 - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{\|\mathbf{r}_i - \mathbf{R}_A\|} + \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{\|\mathbf{r}_i - \mathbf{r}_j\|} \quad (1c)$$

where  $\Psi$  is the  $N$ -particle wave function;  $\mathbf{H}$  is the electronic Hamiltonian operator which includes terms for both the potential and kinetic energy;  $\varepsilon$  is the energy of the system.

Numerous schemes emerged on computing the needed quantities; among them is a well known Hartree-Fock approach [18]. The Hartree-Fock method is very computationally intensive and not practical beyond few metal atoms; scales on the order of  $N^4$ , where  $N$  is

the dimension of the problem, and for a more precise calculation can become as high as  $N^8$  according to the perturbation theory [19]. Density functional theory (DFT) [4] [5] emerged as a more feasible alternative; scales on the order of  $N^3$ . DFT became a method of choice for describing the ground state properties of metals, semiconductors, and insulators. One of the largest known ab-initio simulations by means of DFT have been for nonmetallic structures with up to 1,500 atoms [7]. Anitescu et al. [8] proposed a theoretical framework for the investigation of these nanostructures at a fraction of the cost which targets simulations of nanostructures comprised of hundreds of thousands of atoms are possible.

The ground framework of the nanostructure methodology proposed by Anitescu et al. has been implemented using C++. The goal of this project is to implement an object-oriented framework that will provide testing mechanisms of the evolving code. The byproduct of the testing effort will also be to design modules for the generation of the new crystallographic structures as an input. For an output, the design of the module provides an output that is compliant with visualization software, such as TECPLOT, providing simple rotating, zooming and slicing facilities of the 3D electronic mesh and atomic structure. The ground framework of the module also supports animation for the iterative optimization approach.

## GENERAL FRAMEWORK

Although the methodology utilizes a two-step approach, the two problems are independent of each other according to the Born-Oppenheimer assumption [6]. When the electronic problem is solved, the positions of the nuclei from the ionic problem are obtained. With the positions of the nuclei fixed, the electronic density distribution is first computed. The solution of the electronic problem then becomes input for the ionic problem during which the positions of nuclei are changed to minimize the total energy associated with the nanostructure, resulting in an iterative approach.

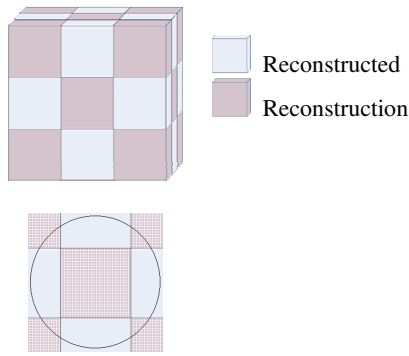


Figure 1: Electronic Problem.

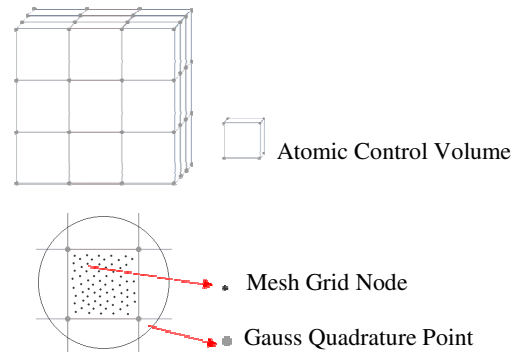


Figure 2: Ionic Problem.

### Electronic Problem

The electronic problem focuses on calculating the electronic density of the nanostructure. Having the positions of the nuclei, the electronic density is calculated in some of the domains, depicted in purple (figure 1). Each reconstruction sub domain is meshed and the electronic density is calculated at each node. Using 3D interpolation over the reconstruction domains, the electronic density in the rest of the domain is obtained. The interpolation takes into account the deformation of the structure through a deformation

mapping. Since the value of the density in the reconstruction subdomains depends on the value of the electron density in the reconstructed sub domains a self consistency loop is employed.

### Ionic Problem

The main focus of the ionic problem is to calculate the most stable shape of the crystal structure. Namely, to calculate the equilibrium configuration of a nanostructure, which is provided by the distribution of the nuclei that minimizes the energy. The whole domain is subdivided into atomic control volumes (figure 2). Within each atomic control volume (ACV), none or thousands of nuclei may reside. For example, if the atomic control volume is cubic, each nucleus inside the ACV is represented by a set of 8 nodes, which are referred to as control points. Only control points are used to impose equilibrium conditions. Following the quasi-continuum methodology [20], the position of each nucleus is determined relative to the position of control points via shape functions.

## **MATERIALS AND METHOD**

The code for the object-oriented nanostructure investigation was written in C++. It can be compiled using either MS Visual Studio C++ .NET (7.1) or GNU C++ compiler (GCC/G++ 3.4.4). Build tools have been developed i.e. make/batch files, for both Linux and Windows platforms. It can be interfaced to any third party optimization software capable of solving bound constrained minimization problems being provided with the function gradient, initial and a penalty value. Other third party software that is required to

run the code on a Linux platform includes TAO [9], PETSc [10] and MPICH2 [11]. On a windows platform, Cygwin [12] must be installed to emulate the Linux environment.

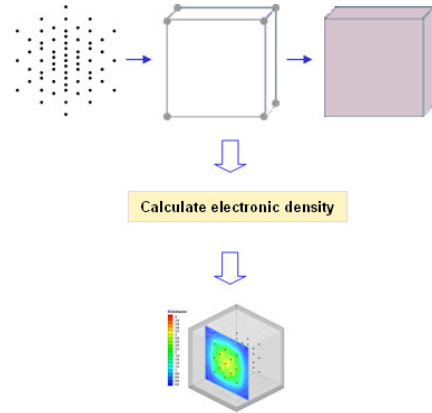


Figure 3: Nanostructure investigation code structure.

Currently, the structure of the code is divided into four stages: preprocessing, electronic problem, ionic problem and post processing (figure 3). In the preprocessing stage, a nanostructure simulation module is defined. A crystal structure of interest is defined by the user. A cubic atomic control volume is then automatically generated for the quasicontinuum approach. Then the crystal structure is then embedded inside the electronic uniform mesh. Next, the initial guess for the electronic density is provided. This could be a uniform distribution throughout the electronic problem. Finally, the deformation map is initialized to be the identity mapping. Once the nanostructure simulation module is defined, the electronic density distribution is then calculated. The electronic problem can be solved both internally and externally. While not yet supported, when solving externally, a specialized code such as NWChem or Gaussian03 can be utilized. Internally, instead of solving the system of integral equations for a given electronic density and the explicit form for the energy functional, using TAO one large

optimization problem is solved. Independent of the type of solver invoked, the electronic density is the solution of the following optimization problem:

$$\min_{\rho} E[\rho; \rho_A] \quad (2a)$$

$$\int \rho(\mathbf{r}) d\mathbf{r} = N \quad (2b)$$

where  $E$  is the energy,  $\rho$  is the nuclear density, and  $N$  is the number of electrons present. Within the ionic problem, only the quasicontinuum component through the atomic control volumes has been implemented so far. In the postprocessing stage, output is generated for both the electronic and the crystal structure in two independent data files: mesh zones and crystal zones. Contour plots can be generated using third party visualization software, i.e. TECPLOT or MATLAB, to view the electronic distribution.

### API Interfaces

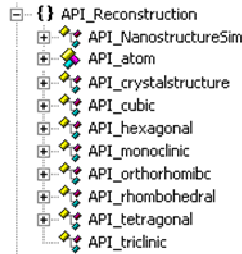


Figure 4. Nanostructure investigation API interfaces.

A special interface is implemented in nanostructure investigation code within the *API\_Reconstruction* namespace (figure 4). The interface's object-oriented design encourages encapsulation of data and simplicity of use. A user is able to initialize all the parameters required for running the simulation code using third party optimization software, without having to understand or being exposed to the inner structure of the code. The *API\_NanostructureSim* is the main class providing the required interface to TAO. The constructor of this class requires the size of the electronic mesh, number of



electronic density cells in all x-, y-, z-directions, and lastly a collection of atoms in the nanostructure. The *API\_crystalstructure* enables a user to define a custom heterogeneous crystal structure using the *API\_atom* class where each atom may be assigned a different atomic number representing a given element. An *API\_atom* is defined by location in space with its corresponding charge. There is also an option to create a bravais lattice; 14 different bravais lattices are supported (table 5). A crystal structure may also be initialized via the *element* namespace, where the crystal structure and all the element parameters are already predefined based on the chosen element. Within the *API\_Reconstruction* namespace it is possible to extend the interface. If a particular feature is in need, it is very straightforward to provide an interface for it.

### Bravais Lattices

Crystalline lattices are distinct lattice types constituted by a regular array of identical units, periodically repeated in space [1]. In 1850, Auguste Bravais proposed a total of fourteen bravais lattices obeying the following criteria: unit cell is the repeating unit in the crystal, opposite faces of a unit cell are parallel and edge of the unit cell connects equivalent points. Bravais lattices can be grouped into seven crystal systems; cubic, tetragonal, orthorhombic, hexagonal, rhombohedral, monoclinic and triclinic. Within each crystal system, different possible types may exist; primitive (P), body-centered (I), 2-face-centered (C) and face-centered (F). Classification of the fourteen bravais lattices into the seven crystal systems is shown. Each crystal system and type is described by different lattice constants  $a, b, c$  and angles  $\alpha, \beta, \gamma$ .

Assuming the following convention:

$a$  is the magnitude of  $a_1$

$b$  is the magnitude of  $a_2$

$c$  is the magnitude of  $a_3$

$\alpha$  is the angle between  $a_2$  and  $a_3$

$\beta$  is the angle between  $a_1$  and  $a_3$

$\gamma$  is the angle between  $a_1$  and  $a_2$

These lattices can be generated using three primitive translation vectors [2]. The equilibrium atomic positions coincide with the linear combination of the bravais translation vectors  $a_1$ ,  $a_2$  and  $a_3$ .

$$R_n = n_1 a_1 + n_2 a_2 + n_3 a_3 \quad (3)$$

Appendix A defines primitive translation vectors for all bravais lattices.














Cubic [ P I F ]				<ul style="list-style-type: none"> <li>• <math>a = b = c</math></li> <li>• <math>\alpha = \beta = \gamma = 90^\circ</math></li> </ul>
Tetragonal [ P I ]				<ul style="list-style-type: none"> <li>• <math>a = b \neq c</math></li> <li>• <math>\alpha = \beta = \gamma = 90^\circ</math></li> </ul>
Orthorhombic [ P I C F ]				<ul style="list-style-type: none"> <li>• <math>a \neq b \neq c</math></li> <li>• <math>\alpha = \beta = \gamma = 90^\circ</math></li> </ul>
Hexagonal				<ul style="list-style-type: none"> <li>• <math>a = b \neq c</math></li> <li>• <math>\alpha = \beta = 90^\circ</math></li> <li>• <math>\gamma = 120^\circ</math></li> </ul>
Rhombohedral				<ul style="list-style-type: none"> <li>• <math>a = b = c</math></li> <li>• <math>\alpha = \beta = \gamma \neq 90^\circ</math></li> </ul>
Monoclinic [ P C ]				<ul style="list-style-type: none"> <li>• <math>a \neq b \neq c</math></li> <li>• <math>\alpha = \gamma = 90^\circ</math></li> <li>• <math>\beta \neq 90^\circ</math></li> </ul>
Triclinic				<ul style="list-style-type: none"> <li>• <math>a \neq b \neq c</math></li> <li>• <math>\alpha \neq \beta \neq \gamma \neq 90^\circ</math></li> </ul>

Table 1: 14 Bravais Lattices

## RESULTS

One of the first 3D verification tests was the simulation of a single Hydrogen atom and 4096 electronic density mesh elements (figure 5). The test ran on a single 2 GHz Mobile Intel Pentium M processor running XP Professional, compiled using GNU C++ compiler. Simulation took 4 minutes and 20 seconds carrying out 27 iterations using TAO for the electronic density to converge with the penalty tolerance of 0.0001. Simulation of a cubic primitive unit cell (figure 6) comprised of 8 atoms and 6859 mesh nodes ran for 14 hours and 12 minutes carrying out 54 iterations to converge. The same simulation was run on a fine mesh but with a smaller electronic density mesh domain. Although the simulation ran longer, the results unexpectedly degraded due to the artifacts that emerged at the boundaries (figure 7). In order to obtain good results, it is crucial to define a large enough discretization domain. Small domains lead to artifacts close to the boundaries.

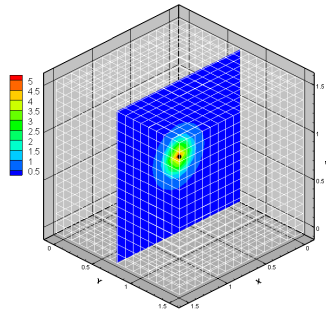


Figure 5: Single atom simulation.

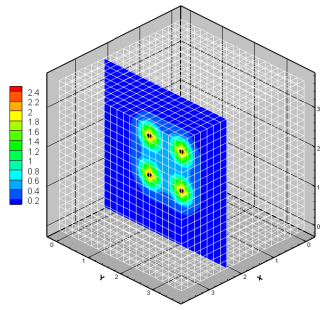


Figure 6: Cubic primitive, unit cell simulation (coarse).

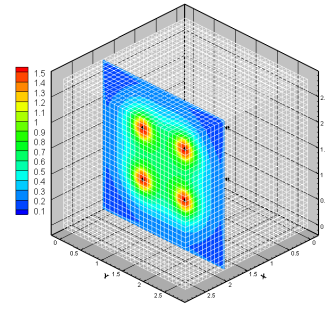


Figure 7: Cubic primitive, unit cell simulation (fine).

One of the bigger simulations that were run took about 14 hours to complete and 12 iterations with the penalty tolerance of 0.001. The simulated crystal structure was a cubic face-centered comprised of 63 crystal atoms and 59319 mesh elements. The electronic

density mesh domain was not as big as it should have been, resulting in artifacts at the boundaries (figure 8).

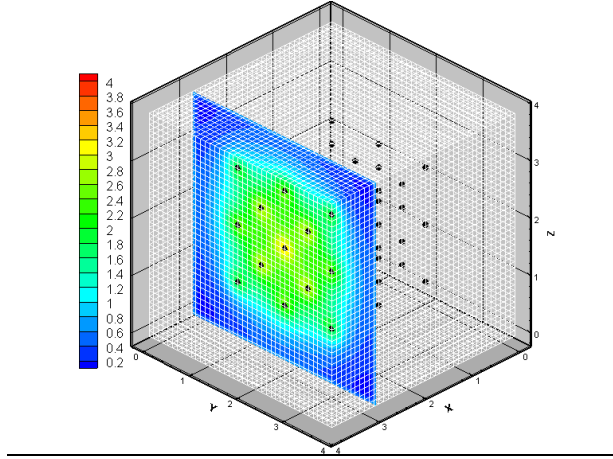


Figure 8: Cubic face-centered, 2 unit cells in x-, y-, and z-directions simulation.

## DISCUSSION AND CONCLUSION

At present, there is still a great deal of work that must be done. First, the code should include automatic mesh refinement of the electronic density mesh, for the electronic problem. This will save a tremendous amount of computing time. As it can be seen in figure 6, computing the electronic density with the same accuracy inside the whole domain is inefficient. A fine mesh around the nuclei and a coarse mesh everywhere else are essential. Next in order is implementing an interface for use of any higher order shape function in the ionic problem. Following the quasi-continuum methodology, since the positions of a nucleus are determined relative to the position of control points using shape functions; it is important to specify a custom shape function. It is also vital to parallelize the main subroutines of the code using MPICH2. This will significantly speed up the

computation time. Another goal is to extend the modeling capabilities by introducing more elaborate energy functionals and more sophisticated DFT models for computing the electronic density. This is an active area of research in the quantum chemistry community.

## **ACKNOWLEDGEMENTS**

The effort described in this paper was preformed at the Mathematics and Computer Science Division, a part of Argonne National Lab in Argonne, Illinois. I would like to take this opportunity to thank the Department of Energy and Office of Science for providing me with the opportunity to participate in the exceptional Science Undergraduate Laboratory Internship program. I would also like to express my sincere gratitude to my mentors Mihai Anitescu and Dan Negrut for the assistance and guidance that they have provided throughout my appointment.

## REFERENCES

- [1] W. Kohn, L. J. Sham, Self-consistent equations including exchange and correlation effects. *Physics Review*, 140A 1133–A1138, 1965.
- [2] G. Burns and A. Glazer, *Space Groups for Solid State Scientists*, Academic Press 1978.
- [3] G. Grosso and G. P. Parravicini, *Solid State Physics*, Cambridge, University Press, 2000.
- [4] P. Hohenberg and W. Kohn, *Physics Review* 136B 864, 1964.
- [5] W. Kohn and L. J. Sham, *Physics Review* 140A 1133, 1965.
- [6] M. Born and K. Huang, *Dynamical Theory of Crystal Lattices*, Oxford Univ. Press, Clarendon, 1954.
- [7] C. K. Skylaris, P. D. Haynes, A. A. Mostofi, M. C. Payne, Introducing ONETEP: Linear-scaling density functional simulations on parallel computers, *The Journal of Chemical Physics*, 122(8), 084119, 2005.
- [8] M. Anitescu, D. Negrut, T. Munson, P. Zapol, *Density Functional Theory-Based Nanostructure Investigation: Theoretical Considerations*, (in press).
- [9] TAO, Toolkit for Advanced Optimization, <http://www-unix.mcs.anl.gov/tao/>.
- [10] PETSc, Portable, Extensible Toolkit for Scientific Computation, <http://www-unix.mcs.anl.gov/petsc/petsc-2/>.
- [11] MPICH2, Message-Passing Interface (MPI), <http://www-unix.mcs.anl.gov/mpi/mpich2/>.
- [12] CYGWIN, Linux-like environment for Windows, <http://www.cygwin.com/>.
- [13] Nanoparticles and the environment. *Reviews in Mineralogy and Geochemistry*, 2001.
- [14] S. A. Chambers, S. Thevuthasan, R. F. C. Farrow, R. F. Marks, J. U. Thiele, L. Folks, M. G. Samant, A. J. Kellock, N. Ruzyski, D. L. Ederer, and U. Diebold, Epitaxial growth and properties of ferromagnetic co-doped TiO<sub>2</sub> anatase, *Applied Physics Letters*, 79, 3467-3469, 2001.

- [15] S. Azad, O. A. Marina, C. M. Wang, L. Saraf, V. Shutthanandan, D. E. McCready, A. El-Azab, J. E. Jaffe, M. Engelhard, C. Peden, and S. Thevuthasan, Nanoscale effects on ion conductance of layer-by-layer structures of gadolinia-doped ceria and zirconia, *Applied Physics Letters*, (to appear).
- [16] J. H. Sun, W.-H. Fan, Y. Xu, D. Wu, and Y.-H. Sun, Nano-sized SiO<sub>2</sub> sol-gel for structure-controlled optical coatings, *Molecular Crystals and Liquid Crystals*, 337, 85-88, 1999.
- [17] A. Perera and J. Krusius, All-level electron-beam lithography for trench isolated nano-metal-oxide semiconductor devices, *Journal of Vacuum Science and Technology B*, 8, 1343-1347, 1990.
- [18] A. Szabo and N. Ostlund, *Modern Quantum Chemistry*, Dover, 1989.
- [19] C. Møller and M. S. Plesset, *Phys. Rev.* 46, 618 (1934).
- [20] E. Tadmor, M. Ortiz, and R. Phillips, Quasicontinuum analysis of defects in solids, *Philosophical Magazine A*, 73, 1529–1563, 1996.

## APPENDIX

### A. Primitive vectors for bravais lattices.

#### Cubic

Primitive

$$a_1 = a \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad a_2 = b \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad a_3 = c \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Body-centered

$$a_1 = \frac{a}{2} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \quad a_2 = \frac{b}{2} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \quad a_3 = \frac{c}{2} \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

Face-centered

$$a_1 = \frac{a}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \quad a_2 = \frac{b}{2} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad a_3 = \frac{c}{2} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

#### Tetragonal

Primitive

Same as cubic primitive

Body-centered

$$a_1 = \frac{1}{2} \begin{bmatrix} a \\ -b \\ c \end{bmatrix} \quad a_2 = \frac{1}{2} \begin{bmatrix} a \\ b \\ -c \end{bmatrix} \quad a_3 = \frac{1}{2} \begin{bmatrix} -a \\ b \\ c \end{bmatrix}$$

#### Orthorhombic

Primitive

Same as cubic primitive

2-Face-centered

$$a_1 = \frac{1}{2} \begin{bmatrix} a \\ -b \\ 0 \end{bmatrix} \quad a_2 = \frac{1}{2} \begin{bmatrix} a \\ b \\ 0 \end{bmatrix} \quad a_3 = \frac{1}{2} \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

Body-centered

Same as tetragonal body-centered

Face-centered

$$a_1 = \frac{1}{2} \begin{bmatrix} a \\ 0 \\ c \end{bmatrix} \quad a_2 = \frac{1}{2} \begin{bmatrix} a \\ b \\ 0 \end{bmatrix} \quad a_3 = \frac{1}{2} \begin{bmatrix} 0 \\ b \\ c \end{bmatrix}$$

#### Hexagonal



$$a_1 = a \begin{bmatrix} \sqrt{\frac{3}{4}} \\ -\frac{1}{2} \\ 0 \end{bmatrix} \quad a_2 = b \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad a_3 = c \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Rhombohedral

$$a_1 = a \begin{bmatrix} \sqrt{\frac{2}{3}}(1 - \cos \gamma) \\ 0 \\ \sqrt{1 - \frac{2}{3}}(1 - \cos \gamma) \end{bmatrix} \quad a_2 = b \begin{bmatrix} -\frac{1}{2}\sqrt{\frac{2}{3}}(1 - \cos \gamma) \\ \sqrt{\frac{1}{2}}(1 - \cos \gamma) \\ \sqrt{1 - \frac{2}{3}}(1 - \cos \gamma) \end{bmatrix} \quad a_3 = c \begin{bmatrix} -\frac{1}{2}\sqrt{\frac{2}{3}}(1 - \cos \gamma) \\ -\sqrt{\frac{1}{2}}(1 - \cos \gamma) \\ \sqrt{1 - \frac{2}{3}}(1 - \cos \gamma) \end{bmatrix}$$

Monoclinic

Primitive

Case  $\gamma = \beta = 90^\circ$

$$a_1 = a \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad a_2 = b \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad a_3 = c \begin{bmatrix} 0 \\ \cos \alpha \\ \sin \alpha \end{bmatrix}$$

Case  $\gamma = \alpha = 90^\circ$

$$a_1 = a \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad a_2 = b \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad a_3 = c \begin{bmatrix} \cos \alpha \\ 0 \\ \sin \alpha \end{bmatrix}$$

Case  $\beta = \alpha = 90^\circ$

$$a_1 = a \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad a_2 = b \begin{bmatrix} \cos \gamma \\ \sin \gamma \\ 0 \end{bmatrix} \quad a_3 = c \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

2-Face-centered

Case  $\gamma = \beta = 90^\circ$

$$a_1 = a \begin{bmatrix} \frac{1}{2} \\ -\frac{\cos \alpha}{2} \\ -\frac{\sin \alpha}{2} \end{bmatrix} \quad a_2 = b \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad a_3 = c \begin{bmatrix} \frac{1}{2} \\ \frac{\cos \alpha}{2} \\ \frac{\sin \alpha}{2} \end{bmatrix}$$

Case  $\gamma = \alpha = 90^\circ$

$$a_1 = a \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 0 \end{bmatrix} \quad a_2 = b \begin{bmatrix} -\frac{1}{2} \\ 2 \\ 1 \\ 2 \\ 0 \end{bmatrix} \quad a_3 = c \begin{bmatrix} \cos \beta \\ 0 \\ \sin \beta \end{bmatrix}$$

Case  $\beta = \alpha = 90^\circ$

$$a_1 = a \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad a_2 = b \begin{bmatrix} \frac{\cos \gamma}{2} \\ \frac{\sin \gamma}{2} \\ 1 \\ 1 \\ 2 \end{bmatrix} \quad a_3 = c \begin{bmatrix} -\frac{\cos \gamma}{2} \\ \frac{\sin \gamma}{2} \\ 1 \\ 1 \\ 2 \end{bmatrix}$$

Triclinic

$$a_1 = a \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad a_2 = b \begin{bmatrix} \cos \gamma \\ \sin \gamma \\ 0 \end{bmatrix} \quad a_3 = c \begin{bmatrix} \frac{\cos \beta}{\cos \alpha - \cos \beta \cos \gamma} \\ \frac{\sin \gamma}{\sqrt{1 - \cos^2 \gamma - \cos^2 \beta - \cos^2 \alpha + \cos \gamma + \cos \beta + \cos \alpha}} \\ \frac{\sin \gamma}{\sin \gamma} \end{bmatrix}$$